September 1986 LIDS-P-1601

SOME ISSUES IN DISTRIBUTED ASYNCHRONOUS ROUTING IN VIRTUAL CIRCUIT DATA NETWORKS.[*]

Wei K. Tsai, John N. Tsitsiklis, Dimitri P. Bertsekas

Dept. of Electrical Engineering and Computer Science
Laboratory for Information and Decision Systems
M.I.T.
Cambridge, MA 02139

ABSTRACT

We consider the behavior of distributed asynchronous routing algorithms for optimizing the flows in a virtual circuit data network, with respect to a given convex cost function. The algorithms operate with minimal synchronization of computations and information exchange between different processors and consist of gradient projection iterations which compute a target set of flows for each path. Then, the processors try to make the actual flows equal to the target flows, by appropriately assigning paths to incoming, new virtual circuits. We concentrate on the "many small users" case, in which there is (on the average) a very large number of virtual circuits, each one requiring a small communication rate. This note is a followup to our earlier paper [TsBe] and addresses the limiting behavior when the frequency of iteration becomes infinite relative to the frequency of information exchange between nodes.

I. MODEL DESCRIPTION.

We are given a network described by a directed graph $G = (V, E)$. (V is the set of nodes, E is the set of directed links.) For each pair $w = (i, j)$ of distinct nodes i, j, (also called an origin-destination, or OD pair), let P_w be a given set of paths from i to j containing no loops. (These are the candidate paths to which virtual circuits will be assigned for transmitting messages from i to j.) For each OD pair w and for any time t, there will be a total of $N_w(t)$ active virtual circuits linking node i to node j. These virtual circuits are assigned to paths, $N_p(t)$ being assigned to path p. (Thus, $N_w(t) = \sum_{p\in P_w} N_p(t)$.) New virtual circuits for OD pair w are generated according to a Poisson process, at a rate λ_w/ϵ, where ϵ is a small positive parameter. Newly generated virtual circuits are assigned to a path $p \in P_w$ and remain assigned to that path during the entire lifetime of the virtual circuit, which is an (independent) exponential random variable with rate μ_w. Each virtual circuit for OD pair w is assumed to require communication rate ϵ from each link in the path to which it is assigned. (Thus, by letting ϵ be very small, we are at the "many small users" situation.)

We are primarily interested in the case where ϵ is very small and will therefore consider the asymptotic performance of routing schemes, as $\epsilon \to 0$. For this reason, we prefer to work with the variables $r_w(t)$ and $x_p(t)$ defined by $\epsilon N_w(t)$, $\epsilon N_p(t)$, respectively. Notice that the mean of $r_w(t)$ (at steady state) is equal to $r_w = \lambda_w/\mu_w$ and is therefore independent of ϵ. For any link (i,j), we also define the flow $F_{ij}(t)$ through that link to be equal to the sum of $x_p(t)$, over all paths p which use link (i,j).

We introduce a cost function $\bar{D}$ which is meant to penalize congestion (large flows) through each link. In particular, we assume the separable form

$$\bar{D}(t) = \sum_{(i,j)\in E} \bar{D}_{ij}(F_{ij}(t)).$$

We assume that each $\bar{D}_{ij}$ is twice continuously differentiable and its second derivative is bounded away of zero. (In particular, $\bar{D}_{ij}$ is strictly convex.) Since the F_{ij}'s are functions of the x_p's, we may rewrite the cost function in terms of the x_p variables to obtain the form $D(x) = \sum_{(i,j)\in E} D_{ij}(x)$, where x is the vector of all x_p's. Clearly, each D_{ij} inherits the convexity property of the $\bar{D}_{ij}$'s.

We are interested in routing schemes which minimize $\limsup_{t\to\infty} E[D(x(t))]$. Let D^* be the minimum value of $D(\cdot)$ subject to the constraints that $x_p \geq 0$, $\sum_{p\in P_w} x_p = r_w$, $\forall w$, $\forall p \in P_w$. As $\epsilon \to 0$, $t \to \infty$, the random variable $r_w(t)$ converges to its steady state mean r_w in the mean square and this fact may be exploited to show that there exist routing policies under which $\lim_{\epsilon\to 0} \limsup_{t\to\infty} E[D(x(t))] \leq D^*$. On the other hand, $\limsup_{t\to\infty} E[D(x(t))] \geq D^*$, for every routing scheme, this being a consequence of Jensen's inequality. Therefore, an algorithm for the deterministic multicommodity network flow problem (for which D^* is the optimal value) may be used as a guide for obtaining an asymptotically (as $\epsilon \to 0$) optimal routing scheme.

The particular scheme we propose is based on the gradient projection algorithm for the above defined multicommodity flow problem, which consists of the following iteration (for each w): Let $x_w = (x_p;\ p \in P_w)$ be the vector of all path flows for a given OD pair w. Let $\bar{p}$ be a path in P_w with the smallest value of $(\partial D/\partial x_{\bar{p}})(x_w)$. We then let

$$x_p \leftarrow \max\{0, x_p - \mu_p\gamma(\frac{\partial D}{\partial x_p}(x_w) - \frac{\partial D}{\partial x_{\bar{p}}}(x_w))\}, \quad p \neq \bar{p} \quad (1.a)$$

$$x_{\bar{p}} \leftarrow (r_w - \sum_{p\neq\bar{p},\ p\in P_w} x_p). \quad (1.b)$$

Here μ_p is a positive scaling constant, typically obtained from an approximation of the matrix of second derivatives of D), and γ is a small positive stepsize.

In a realistic data network as decsribed above, iteration (1) cannot be implemented exactly if x_w is to stand for the vector of actual flows through the paths $p \in P_w$. Some of the reasons are the following: a) Due to the stochastic nature of the generation and extinction of virtual circuits, it is impossible to enforce a desired number of them, for each p; b) The processor who is to execute the iteration (1) for a certain OD pair w may not have access to the exact current value of the derivative of D, evaluated at the vector x of current path flows; c) The value of r_w may not be known exactly; in fact in realistic situations r_w varies slowly with time and a good routing algorithm should be able to track such changes without causing flows to be far from optimal.

[*] Research supported by the National Science Foundation under grant ECS-8217668 and by a Presidential Young Investigator Award.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **SEP 1986**	2. REPORT TYPE	3. DATES COVERED **00-09-1986 to 00-09-1986**

4. TITLE AND SUBTITLE **Some Issues in Distributed Asynchronous Routing in Virtual Circuit Data Networks**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Massachusetts Institute of Technology,Laboratory for Information and Decision Systems,77 Massachusetts Avenue,Cambridge,MA,02139-4307**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**3**	

In the next paragraph we describe how iteration (1) would be implemented in a realistic environment so as to overcome the above mentioned difficulties.

We assume that the processor in charge of the OD pair w (PR_w, for short) has available at each time t a target flow $\bar{x}_p(t)$ for each path $p \in P_w$. (We denote by $\bar{x}_w(t)$ the vector with components $\bar{x}_p(t)$, $p \in P_w$.) These target flows are updated at times t_w^n, $n = 1, 2, \ldots$. We assume that for some scalars δ, Δ,

$$0 \le \delta \le t_w^{n+1} - t_w^n \le \Delta, \qquad \forall n, w. \tag{2}$$

Other than the above inequalities, we do not impose any restrictions on the sequence $\{t_w^n\}$, thus assuming minimal synchronization of the computations of processors in charge of different OD pairs. Suppose that at time t_w^n, processor PR_w has available estimates λ_p^n of partial derivatives $\partial D/\partial x_p$, for each $p \in P_w$, evaluated at $x(t_w^n)$. This processor is typically the origin node for OD pair w and we may reasonably assume that it has available $r_w(t)$ and $x_w(t)$, at each time t. Then, this processor evaluates a vector $\bar{x}_w^n$ of target flows using the formulas

$$\bar{p}_n = \arg\min\{\lambda_p^n\}, \tag{3.a}$$

$$\bar{x}_p^n = \max\{0, x_p(t_w^n) - \gamma\mu_p^n(\lambda_p^n - \lambda_{\bar{p}_n}^n)\}, \quad p \neq \bar{p}_n, \tag{3.b}$$

$$\bar{x}_{\bar{p}_n}^n = r_w(t_w^n) - \sum_{p \neq \bar{p}_n,\ p \in P_w} x_p(t_w^n). \tag{3.c}$$

We assume that there exist constants $m, M > 0$ such that $0 < m < \mu_p^n < M$.

The estimates λ_p^n are assumed to be formed as follows: a processor observing link (i,j) computes, once in a while, the derivative of $\bar{D}_{ij}$, evaluated at $F_{ij}(t)$, where t is the current time and transmits this value to processor PR_w. Such derivatives (obtained from each link (i,j)) are used by processor PR_w to construct the estimate λ_p^n as the sum of $\bar{D}'_{ij}$ over all links (i,j) on path p. This estimate would be exact if $F_{ij}(t) = F_{ij}(t_w^n)$; however, due to lack of synchronization between processors and communication delays this will not be the case in general. Nevertheless, if the flows in the network change slowly, this estimate will be fairly accurate. (The above described scheme may be generalized by allowing the processor associated with link (i,j) to use a short term average of F_{ij}, rather the instantaneous value $F_{ij}(t)$ at a single time t.) We assume that the processors associated with each link evaluate the appropriate derivatives at least once every B time units, where B is some constant; furthermore, communication delays are also assumed to be bounded by B.

It remains to describe how processor PR_w assigns incoming virtual circuits to paths, during the time interval $[t_w^n, t_w^{n+1})$. The objective is of course to make the actual flows $x_p(t)$ as close as possible to the target flows $\bar{x}_p^n$, but there are several alternatives; we present two:

(i) **Randomization:** Each incoming virtual circuit is assigned to path p with probability $\bar{x}_p^n / r_w(t_w^n)$ and independently from other assignments.

(ii) **Metering:** A new virtual circuit generated at time t is assigned to the path p for which the value of $\bar{x}_p^n - x_p(t)$ is largest. (Ties are broken by randomization.)

Metering is generally recommended over randomization in practice since it tends to bring about more quickly a close match between target and actual path flows. On the other hand, the use of metering has an interesting adverse effect on convergence when the frequency of iteration is very large relative to the frequency of information exchange. This is the main point of this note, and is discussed in the next section.

Related Research: The above described scheme is motivated from an algorithm implemented in the CODEX network (see [BeGa1], Section 5.8). The application of nonlinear programming methods for distributed routing in data networks goes back to [Ga]. The particular gradient projection method considered here was proposed in [Be], [BeGa2]. The asynchronous version of the gradient projection algorithm was analyzed in [TsBe], following more general studies of asynchronous descent algorithms [TsBeAt]. In all these references, the stochastic nature and short–term variations of the input traffic are ignored. In [GaBe] the "many small users" assumption is introduced in a stochastic setting and asymptotic optimality (as $\epsilon \to 0$) is proved for a different class of routing algorithms, under the assumption of synchronism. Finally [Ts] considered the simultaneous effects of asynchronism and the stochastic nature of input traffic (under the many small users assumption) thus integrating previous models and approaches. The statements made in the next section are variations of some of the results in [Ts].

II. DISCUSSION OF PERFORMANCE.

We discuss separately two cases:

A: Let us assume that the constant δ of inequality (2) is nonzero. Equivalently, there is a lower bound on the time between consecutive updates of the target flows, by each processor. Then, the following result has been proved in somewhat different form in [Ts]: For any fixed positive value of δ we may choose the stepsize γ small enough and guarantee that, with either randomization or metering, asymptotic optimality is obtained, in the sense that $\lim_{\epsilon \to 0} \limsup_{t \to \infty} E[D(x(t))] = D^*$.

The proof of the above statement is quite long because of the technicalities involved. However, the outline of the argument is fairly simple. We first discard the possibility that $|r_w(t) - r_w|$ is not negligible, this being a low probability event, for t large and ϵ small. Then, we choose γ to be, say, an order of magnitude smaller than δ. Then, the difference $\bar{x}_p^n - x_p(t_w^n)$ is very small when compared with the length of the time interval $[t_w^n, t_w^{n+1}]$. Thus, by the end of that time interval, $x_p(t)$ is equal to $\bar{x}_p^n$ plus some random deviation which is negligible as $\epsilon \to 0$. Thus, after neglecting small random deviations, we may safely forget the stochastic nature of the generation and extinction of virtual circuits. We are therefore in the setting of a deterministic asynchronous gradient projection algorithm, whose convergence to an optimum of the cost function can be proved using the techniques of [TsBe, TsBeAt].

B: We now consider the case where $\delta = 0$. This means that it is possible that one processor performs an unbounded number of iterations before other processors get a chance of performing a single iteration. For this case, it can be shown by means of an example that if metering is employed and no matter how γ is chosen, the algorithm may be non–convergent. The essence of such an example admits a simple explanation. For asynchronous gradient–like algorithms to be convergent one generally needs that the difference between true derivatives and estimates of derivatives, used in the computation, is of the order of the stepsize γ. This requires that the flows at the time t_w^n of an iteration are not substantially different from the flows that were used in the evaluation of the derivative estimate λ_w^n. This in turn requires that $F_{ij}(t) - F_{ij}(s)$ be of the order of γ, when $t - s$ is of the order of B (which is the bound on communication delays). However, with metering, even if $\bar{x}_p^n - x_p(t)$ is of the order of γ, still there is always a path flow $x_p(t)$ which changes with $O(1)$ speed. Once $x_p(t)$ reaches $\bar{x}_p^n$, (which takes $O(\gamma)$ time), processor PR_w may compute a new target and the process will be repeated. Thus, some path flows $x_p(t)$ may keep moving at (order of) unit speed for a time interval of $O(1)$ magnitude, before new derivative information is obtained from other processors. It follows that the previously mentioned requirements for convergence fail to hold, derivative estimates have a $O(1)$ error and there is nothing to guarantee that the algorithm progresses in a descent direction.

For randomization, the situation is different: if x_p stands for a generic true flow variable and $\bar{x}_p$ stands for a target variable, then the mean of x_p satisfies a differential equation of the form $dx_p/dt = \mu(\bar{x}_p - x_p)$. Thus if initially $x_p - \bar{x}_p = O(\gamma)$, which is always the case after an iteration, $E[x_p(t)]$ changes with $O(\gamma)$

speed. The variance of $x_p(t)$ goes to zero, as $\epsilon \rightarrow 0$ and it follows that the derivative estimates are wrong only within a $O(\gamma + \epsilon^{1/2})$ factor. By choosing ϵ and γ small enough, the derivative estimates are accurate enough to guarantee that iterations proceed in a descent direction and the techniques of [TsBe] may be used to demonstrate convergence, in a suitable sense. We should mention here that the technique employed in [TsBe] requires that the true flow at the time of an iteration is a convex combination of the true flow at the previous iteration and the target flow computed at that previous iteration. This property is valid in this case, modulo certain stochastic terms which vanish as $\epsilon \rightarrow 0$; this is a straightforward consequence of the differential equation describing the evolution of the mean of x_p, when randomization is employed.

III. REFERENCES.

[Be]D.P. Bertsekas, "A Class of Optimal Routing Algorithms for Communication Networks", in *Proceedings of the 5th International Conference on Computer Communications,* Atlanta, GA, October 1980, pp. 71–76.

[BeGa1]D.P. Bertsekas, R.G. Gallager, *Data Networks,* Prentice Hall, Englewood Cliffs, N.J., 1986.

[BeGa2]D.P. Bertsekas, E.M. Gafni, "Projection Methods for Variational Inequalities with Application to the Traffic Assignment Problem", *Math. Progr. Studies,* Vol. 17, 1982, pp. 139–159.

[Ga]R.G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation", *IEEE Transactions on Communications,* Vol. COM–25, pp. 73–85, 1977.

[GaBe]E.M. Gafni, D.P. Bertsekas, "Asymptotic Optimality of Shortest Path Routing", to appear in the *IEEE Transactions on Information Theory.*

[Ts]W.K. Tsai, "Optimal Quasi–Static Routing for Virtual Circuit Networks Subjected to Stochastic Inputs", Ph.D. Thesis, Department of Electrical Engineering and Computer Science, M.I.T.,1986.

[TsBe]J.N. Tsitsiklis, D.P. Bertsekas, "Distributed Asynchronous Optimal Routing in Data Networks", *IEEE Transactions on Automatic Control,* Vol. AC–31, 4, April 1986, pp. 325–332.

[TsBeAt]J.N. Tsitsiklis, D.P. Bertsekas, M. Athans, "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms", *IEEE Transactions on Automatic Control,* Vol. AC–31, No. 9, 1986, pp. 803–812.